

PATENT

REMARKS

This paper is responsive to a final Office action dated June 17, 2005. Claims 1-26 were examined. Applicant respectfully traverses all of the rejections.

Rejections Under 35 U.S.C. §101

Claims 1-10, 12-23 and 26 are rejected under 35 U.S.C. §101 by the Examiner. In the previous response, Applicant explained that the claims were directed to statutory subject matter. Claims 13 and 23 are method claims that qualify as statutory process claims under MPEP §2106 (IV)(B)(2)(b)(ii). The Office mistakenly characterizes Applicant's arguments as being directed to the "Safe Harbors" of MPEP §2106 (IV)(B)(2)(b)(i). Claim 13 is directed to implementing a parser for an input information encoding and claim 23 is directed to parsing an information encoding. Both claims are statutory process claims per MPEP §2106 (IV)(B)(2)(b)(ii). Claim 26 is directed to an apparatus that can be categorized as a statutory product claim. Claim 1 has been amended to recite "encoded on at least one computer readable medium."

Rejections Under 35 U.S.C. §102 or §103

Claims 1-26 are rejected under 35 U.S.C. §102(b) as being anticipated by or, in the alternative, under 35 U.S.C. §103(a), as obvious over the combination of "Support for Modular Parsing in Software Reengineering" by Peak et al, published in IEEE July 1997 (hereinafter "Peak") in view of the theory of parsing and lexical analysis that one of ordinary skill in the art should know as taught by Aho et al, in "Compilers Principles Tools and Techniques" from September 19, 1985 (hereinafter "Aho"). The Office engages in hedging that suggests the Office acknowledges the inadequacy of Peak alone. However, the addition of Aho still fails to provide any further support for the Office's rejections.

Art of Record Fails to Disclose or Suggest Miniparsers and Office Acknowledges Such

All of the rejections by the Office rely on a faulty foundation assertion by the Office, which is that modular parsers are miniparsers. This mistaken assumption by the Office is relied upon for all of the rejections of the independent claims. In the previous response, Applicant

PATENT

explained that the modular parsers of Peak are implemented for specific grammars (Peak, p. 59, second column, last full paragraph). Peak uses “the terms ‘grammar’ and ‘language model’ as synonyms” (Peak, p. 59, first column, last full paragraph). In contrast, claim 1 recites “wherein respective ones of the miniparsers are limited to particular subsets of syntactic constructs to be parsed in the input information encoding.” To illustrate Peak, a first modular parser of Peak would be for a base language model (grammar GA) and a second modular parser of Peak would parse an extension of the base language model (grammar GB) (see Peak, p. 61, section 4). This point can also be understood by reviewing the bullet points in the Introduction section of Peak at page 58, which refers to the emergence of embedded languages and refers to COBOL having several different implementations. Hence, Peak’s modular parsers handle different grammars. In contrast, dependent claim 8 provides examples of miniparsers. Claim 8 recites “a comment parser; a delimiter parser; a top-level statement parser; a compilation unit parser; and a name parser.” Individual miniparsers handle different subsets of syntactic constructs and not entire grammar or language models, as done by the modular parsers of Peak.

In fact, the Office acknowledges the failure of Peak and Aho to disclose or suggest limiting a miniparser to recognizing only a particular subset of syntactic constructs as recited in independent claim 13, and similarly in claims 23 and 24. Although claim 26 does not recite miniparsers, claim 26 recites “each successive transformation handling only a subset of syntactic constructs in accordance with the grammar.” In response to Applicant asserting the failure of Peak and Aho to disclose or suggest these limitations of the independent claims, the Final Rejection states “Examiner’s response is that this is a good thing” (Final Rejection of 17 June 2005, page 9). Thus, the Examiner clearly acknowledges that the art of record does not disclose or suggest these limitations. However, Applicant refers the Examiner to paragraphs [1034] – [1036] of Applicant’s specification to identify some example benefits of miniparsers as described in Applicant’s specification.

Art of Record Fails to Disclose Successive Operation of Miniparsers or Successive Transformations

The art of record also fails to disclose the successive operation of the miniparsers or successive transformations of an abstract syntax tree. The Office states that the limitations of the

PATENT

independent claims are not an option. The Office then refers to the successive stages of compiling depicted in Aho. Applicant respectfully submits that the Office fails to appreciate the content of Aho and Peak, and fails to examine the claims as a whole or even the particular claim limitations when relying upon Aho's disclosure of successive operations. Claim 1 specifically recites "plural miniparsers each successively operable on a respective abstract syntax tree...that includes transformations of predecessor ones." The Office relies on Peak's disclosure of two modular parsers concatenating possible parses. The modular parsers of Peak do not operate on abstract syntax trees of a predecessor miniparser. The modular parsers operate on their own parser streams and generate possible parses, which are concatenated (see p. 61 of Peak). Recognizing that Peak does not disclose or suggest the limitations of claim 1, the Office refers to Aho. Although Aho discloses successive phases for compiling code as known by anyone of ordinary skill in the art, Aho does not disclose or suggest the successive operations of miniparsers as recited in claim 1. The Office relies heavily on Aho's depiction of a lexical analyzer feeding information into a syntax analyzer on page 13 of Aho. However, there is no depiction of miniparsers operating on successive abstract syntax trees from predecessor miniparsers. Aho only discloses a conventional parser. Applicant refers the Office to Chapter 4 of Aho. This Chapter discloses a parser for syntax analysis. There are no miniparsers or successive transformations that handle only subsets of syntactic constructs. In addition, a lexer or lexical analyzer and syntax analyzer cannot reasonably be construed as miniparsers, especially in light of Aho clearly recognizing a parser as distinct from a lexer. If the Office contends that Peak and Aho disclose or suggest the above discussed limitations of claim 1, Applicant requests that the Office identify where Aho discloses miniparsers each successively operable on a respective abstract syntax tree...that includes transformations of predecessor miniparsers, since Peak discloses concatenating possible parses of modular parsers and fails to disclose the limitations of claim 1.

The Office relies upon these same sections of Aho and Peak to reject claims 13, 23, 24, and 26. However, the same contentions against the rejections of claim 1 apply to the rejections of these independent claims. Claim 13 recites "defining a succession of miniparsers each operable on a respective parse state resulting from a predecessor one of the miniparsers", and claim 24 recites "functional encodings of at least two miniparsers, a first one of the miniparsers executable to transform a first parse tree into a second parse tree and the second one of the

PATENT

miniparsers executable to transform the second parse tree into a third parse tree.” As already stated, neither Aho nor Peak, standing alone or in combination disclose or suggest miniparsers or successive operation of miniparsers as recited in claims 13 and 24. Claim 23 recites “performing plural successive transformations, each successive one of the transformations operating on an abstract syntax tree that is a result of a predecessor one of the transformations,” and claim 26 recites “multipass means for performing plural successive transformations on the encoded information” As stated above, Aho discloses successive phases of a compiler and not successive transformations as recited in claims 23 and 26 and does not disclose each successive transformation being limited to handling only a subset of syntactic constructs.

Dependent claims

As stated above, Aho discloses input flowing between different phases of a compiler, and does not disclose or suggest abstract syntax tree (AST) flowing between different miniparsers. It is also stated above that Peak discloses two parsers operating on separate parse streams and concatenating their resulting possible parses, and a miniparser inputting an AST into another miniparser. Hence, neither Aho nor Peak, standing alone or in combination disclose or suggest “wherein, for at least some of the miniparsers, the respective abstract syntax tree is an output of a respective predecessor one of the miniparsers” as recited in claim 2 and similarly in claim 15, or “wherein, for at least some of the miniparsers, respective input and output abstract syntax trees are separately encoded” as recited in claim 3 and similarly in claim 16.

With regard to claim 6, individual miniparsers handle different subsets of syntactic constructs and not entire grammar or language models, as done by the modular parsers of Peak. Hence, Peak fails to disclose or suggest claim 6, which recites “wherein the syntactic constructs are those defined by a grammar; but wherein none of the miniparsers individually implements the grammar.” The Office states that Peak teaches more than Applicant. However, Applicant contends that the Office fails to appreciate that this is a distinction between miniparsers and modular parsers of Peak.


With regard to claim 8, the Office relies upon the various functionality of a compiler as discloses in Aho. However, claim 8 recites particular miniparsers. The references fail to disclose or suggest miniparsers, and also fail to disclose or suggest the particular miniparsers as

PATENT


recited in claim 8. Similar to claim 8, claim 21 recites "wherein an earlier executed one of the miniparsers associates comment tokens of a compilation unit with respective non-comment tokens; and wherein a later executed one of the miniparsers matches grouping tokens," and claim 22 recites "wherein a still later executed one of the miniparsers segregates tokens into top-level statements." Again, the Office relies upon functionality of a compiler to reject claims 21 and 22. However, the references relied upon by the Office do not disclose or suggest a miniparser behavior as recited in claims 21 and 22.

Neither Aho nor Peak, standing alone or in combination disclose or suggest any of Applicant's claims for at least the reasons given above.

In summary, claims 1-26 are in the case. All claims are believed to be allowable over the art of record, and a Notice of Allowance to that effect is respectfully solicited. Nonetheless, if any issues remain that could be more efficiently handled by telephone, the Examiner is requested to call the undersigned at the number listed below.

<u>CERTIFICATE OF MAILING OR TRANSMISSION</u>	
I hereby certify that, on the date shown below, this correspondence is being	
<input type="checkbox"/> deposited with the US Postal Service with sufficient postage as first class mail and addressed as shown above.	
<input checked="" type="checkbox"/> facsimile transmitted to the US Patent and Trademark Office.	
 Steven R. Gilliam	<u>17-Aug-2005</u> Date
EXPRESS MAIL LABEL: _____	

Respectfully submitted,



Steven R. Gilliam, Reg. No. 51,734
Attorney for Applicant(s)
(512) 338-6320 (direct)
(512) 338-6300 (main)
(512) 338-6301 (fax)